Assignment – 1

1. **Write a java program to find the maximum & minimum element in an array.**

import java.util.Scanner;

class MaxMin {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter how many element you want: ");

        int size = sc.nextInt();

        int[] arr = new int[size];

        System.out.print("Enter the space separated array elements: ");

        for (int i = 0; i < size; i++)

            arr[i] = sc.nextInt();

        int max = Integer.MIN\_VALUE, min = Integer.MAX\_VALUE;

        for (int i = 0; i < size; i++) {

            if (arr[i] > max)

                max = arr[i];

            if (arr[i] < min)

                min = arr[i];

        }

        System.out.println("Max: " + max);

        System.out.println("Min: " + min);

        sc.close();

    }

}

Source Code



user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ javac MaxMin.java

user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ java MaxMin

Enter how many elements you want: 5

Enter the space separated array elements: 2 4 1 6 7

Max: 7

Min: 1

Output

1. **Implement a java program to reverse an array.**

import java.util.Scanner;

class RevArray {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter how many element you want: ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the space separated array elements: ");

        for (int i = 0; i < n; i++)

            arr[i] = sc.nextInt();

        System.out.print("Reversed array: ");

        for (int i = 0; i < n / 2; i++) {

            int temp = arr[i];

            arr[i] = arr[n - i - 1];

            arr[n - i - 1] = temp;

        }

        for (int i = 0; i < n; i++)

            System.out.print(arr[i] + " ");

        sc.close();

    }

}

Source Code



user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ javac RevArray.java

user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ java RevArray

Enter how many element you want: 5

Enter the space separated array elements: 1 2 3 4 5

Reversed array: 5 4 3 2 1

Output

1. **Write a java program to check an array is palindrome or not.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n;

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    int arr[n];

    if (n < 1)

    {

        printf("Invalid input\n");

        exit(0);

    }

    printf("Enter the elements of the array: ");

    for (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    int max = arr[0];

    for (int i = 1; i < n; i++)

    {

        if (arr[i] > max)

        {

            max = arr[i];

        }

    }

    printf("The maximum element in the array is: %d\n", max);

    return 0;

}

Source Code



user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ javac PalindromeArr.java

user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ java PalindromeArr

Enter how many element you want: 4

Enter the space separated array elements: 1 2 2 1

Array is palindrome

user@AnitDesktop MINGW64 /g/My Works/B.Tech-IT/OOPs/java\_ass (main)

$ java PalindromeArr

Enter how many element you want: 4

Enter the space separated array elements: 1 2 3 1

Array is not palindrome

Output

1. **Write a java program using switch case.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n;

    printf("Enter the number of elements in the array: ");

    scanf("%d", &n);

    int arr[n];

    if (n < 1)

    {

        printf("Invalid input\n");

        exit(0);

    }

    printf("Enter the elements of the array: ");

    for (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    int max = arr[0];

    for (int i = 1; i < n; i++)

    {

        if (arr[i] > max)

        {

            max = arr[i];

        }

    }

    printf("The maximum element in the array is: %d\n", max);

    return 0;

}

Source Code



void rotate(int arr[], int n, int pos)

{

    // Adjust position to be within bounds

    if (pos > n)

        pos = pos % n;

    // Create a temporary array to hold the rotated values

    int temp[max];

    for (int i = 0; i < n; i++)

        temp[(i + pos) % n] = arr[i];

    // Step 3: Copy back from temp to arr

    for (int i = 0; i < n; i++)

        arr[i] = temp[i];

}

Source Code: rotate()

()

Enter how many elements you want: 5

Enter the array elements: 1 2 3 4 5

The position of rotation: 3

The array is: 1 2 3 4 5

The rotated array is: 3 4 5 1 2

Output